REMARKS

Claims 1-79 are all the claims pending in the application.

Claim Rejections Under 35 USC § 112 – Second Paragraph

Claim 3. The Examiner rejects claim 3 and requests to point out sections of the specifications and drawings to support the argument. The Applicants respectfully refer the Examiner to paragraph 0071. The Applicants respectfully submit that the word *solely* refers to the fact that after the initial configuration of the system, the system is capable of operating without intervention of the host computer, i.e., between any nodes of the network. Figs. 2 and 3 show the capability of moving data between two networked storage devices 320 through data streamer 200 and without any intervention from the host 310.

Claim 6 is rejected merely due to its dependency on claim 3.

Claim Rejections Under 35 USC § 102(e)

Claims 1–5, 7–9, 19–24, 28, 29, 37, 44–49, 51, 53–56, 78 and 79 are rejected as being anticipated by Roach et al. in US patent 6,314,100 (hereinafter "Roach"). Herein, the Applicants provide some general background information related to a Fiber Channel which is the subject matter of Roach and compare it to a packet based invention which is the subject matter of the present Application.

Roach is related to Fiber Channel networks that are materially different operationally and conceptually from Ethernet or generic networking schemes. In a Fiber Channel, network data is moved using frames, every frame including information that positions the payload of the packet within the higher level protocol construct. This information is placed in the header, see for example Roach column 6, lines 35-45. Specifically, Fiber Channel uses a Sequence procedure to organize the frames while Ethernet and other packet based networks use upper level protocol

(ULP) protocol data unit (PDU). The assembly of Frames into Sequences is significantly simpler than assembly of other networks packets into ULP PDU constructs.

This simpler approach is also used by the Roach system. When it receives unordered frames intervention is required to resolve the order. Roach explains this in relation to steps 520 and 522 of Fig. 5. If a Frame is not the one expected in a sequence, it is sent to the host along with a message that interrupts the host. The host then puts together the Frames into a stream. This leads to the simple conclusion that the system lacks a streamer that is necessary to cause an ordered stream of bytes, if no host intervention is desired.

The present invention hands off to the host a clean and ordered byte stream, with ULP PDU delineation, whether or not packets were received in order or not.

In addition, Fiber Channel has a different topology from a packet network topology due to the fact that Fiber Channel does not deal with data collisions. Therefore there are three types of Fiber Channel topologies used: point-to-point, arbitrated loop, and fabric – none of these resembles the packet network design that allows multiple sources and destinations to reside and communicate simultaneously on the same line. Therefore a person-skilled-in-the-art would not readily use teachings from one communication mechanism to the other due to the significant differences between them. Examiner is urged to refer to the paper titled "Fiber Channel Fundamentals" by Tom Weimar from Spectra Logic Corporation.

Claim 1. Roach does not disclose a packet based network system due to the frame based operation of Fiber Channel systems and the limitation, also shown with respect of Fig. 5 of Roach, where the operation is terminated merely because of an out-of-sequence frame. Such a shortfall cannot be tolerated in a system handling packets and particularly in network systems were packets may arrive in an out-of-order sequence. On the other hand, the data streamer of the present invention is configured to provide the data stream without interruptions to the host because of such problems.

Moreover, nothing in Roach teaches that data is not moved between locations, and even the Examiner points to column 7 lines 37-67 were the use of buffers with available space is

specifically mentioned. That means that data is moved to temporary storage location prior to being moved to a permanent location. By contrast, this is not required according to the present invention as is clearly demonstrated from the elaborate queue and pointer system disclosed. To further clarify this point, the claim is amended to restrict its applicability to packet-based networks.

Claim 2. The claim should be allowed at least based on its dependency.

Claim 3. A close reading of Roach column 1, line 58 through column 2, line 10, including the reference to the PVC does not reveal the reason why the Examiner has decided that this describes a system where the sole activity of the host is for the purpose of system initialization. In Fiber Channel, connections are set up or removed on a continuous basis, however, because of the specific architecture of Fiber Channel this is a non-complex operation, as duly noted by Roach. This has nothing to do with the initialization of the system and Roach makes no reference whatsoever to this issue in this respect. Therefore, claim 3 should be allowed.

Claim 4. The claim should be allowed at least based on its dependency.

Claim 5. The claim should be allowed at least based on its dependency. Further, the Examiner contends that the PVC is a dedicated network communication link but there is nothing to support that notion.. Roach notes that in the Fiber Channel "*There are no complicated permanent virtual circuits (PVCs) to set up.*" (column 2, lines 3-4). It is not understood how the existence or lack thereof of a PVC is equivalent to a dedicated communication link. By virtue of being virtual, the link is not dedicated.

Moreover, Fiber Channel link does not poses nor support the network routing protocols and algorithms required for the necessary level of operation used in the network configuration of the Application.

Claim 7. The claim should be allowed at least based on its dependency. Further, the Applicants request the Examiner to further clarify his position. While Roach discusses LAN and WAN, no teaching in respect of the disclosed invention is made, nor does Roach suggest to connect a LAN or WAN and in fact notes that "*Features of both channels and networks have been incorporated into... Fiber Channel.*" (column 1, lines 41-43). It is not shown at this point that features of a LAN or a WAN have any impact or connection to the Roach system. Neither does it show that a Fiber Channel solution constitutes either a LAN or a WAN. More specifically, in a Fiber Channel point-to-point connectivity it cannot be part of any other network. The arbitrated loop connectivity is limited to connectivity to hard drives. The Fabric connectivity of Fiber Channel allows a point of connection to a LAN or WAN, nonetheless, Fiber Channel is not used as a WAN or LAN due to its well-known limitations.

Claim 8. The claim should be allowed at least based on its dependency. Furhter. The Examiner is believed to be mischaracterizing both the claim and the cited reference. Roach specifically notes that "*A Fiber Channel switch fabric 104 connects... local area networks (LANs) 114. Such LANs include, for example, Ethernet, Token Ring and FDDI networks.*" (column 2, lines 15-19). Nothing in this passage suggest that a Fiber Channel is any kind of a LAN but rather that the Fiber Channel is capable of connecting to them. Specifically, Roach describes in the referenced text (col. 2, lines 10-19) as to how LAN networks like Ethernet Token Ring and FDDI, can be connected to Fiber Channel through a special port in a Fiber Channel switch (see Roach Fig. 1). The connectivity to these networks is therefore not directly to the apparatus that Roach describes and that is contended by the Examiner to be equivalent to the present invention. A Fiber Channel switch fabric is not based on any one of these LANs and hence the argument made by the Examiner is false.

Claim 9. The claim should be allowed at least based on its dependency. Furhter, Roach does not reveal the relief of the Fiber Channel upper layer protocol, also knows as FC-4 (see Roach, column 2, lines 20-27). Further, Roach does not mention nor show that FC-4 would actually ever take place on a host. By contrast, ULP in packet networks typically are preformed on a host and therefore providing a relief to the host from processing all or some of the ULP is

advantageous. Moreover, FC-4 ULP are concerned with the mapping of the Fiber Channel protocol onto protocols such as SCSI and IP, i.e., lower level protocols as far as these types of protocols are concerned. As there is no host to relief Examiner's assertion is believed to be incorrect.

Claim 12. The claim should be allowed at least based on its dependency. Further, this claim has been amended to further clarify the subject matter. The Examiner is to believed to be incorrect. Firstly, a Fiber Channel does not have a network interface which is packet based as discussed in detail above. The only processing node that is referred to with respect to Roach is PENG 408 and it is not configured to handle packet data for the reasons discussed in detail above. There is nothing in the reference provided by the Examiner suggesting an admission and classification unit, and moreover, one would expect that with the reference to Fig. 4 made by the Examiner that a consistent view of a device would be available. Roach does not attempt to do so and the Examiner, merely in hindsight attempts to equate the present invention and Roach.

Examiner then attempts to equate Roach's *manager engine* to *event queue manager* of the present invention is also believed to be incorrect.

Claim 13. The claim should be allowed at least based on its dependency. The reference cited by the Examiner does not suggest an expansion memory to a processing node.

Claim 14. The claim should be allowed for at least being a dependent claim of an allowable claim.

Claim 22. The claim should be allowed at least based on its dependency. Further, the receive and transmit frame buffers (406, 404) of Roach do not constitute the object and application queues explained in great detail in the Application. Nor does Roach teach an even queue manager.

Claims 23. The claim should be allowed at least based on its dependency.. Further, contrary to the Examiner's contention, there is no object queue in Roach and hence it cannot point to a first descriptor.

Claim 24. The claim should be allowed at least based on its dependency. Further, nothing in the reference cited by the Examiner speaks of the communication layers.

Claims 28-29. The claims should be allowed at least based on its dependency. Nothing in the text cited by the Examiner shows a teaching by Roach of an object queue as described in the Application.

Claim 37. The claim should be allowed at least based on its dependency. Further, nothing in the cited reference shows a teaching by Roach with respect to handling of packet data, a handling which is known in the art to be different from that of handling frames of Fiber Channel communication.

Claims 15-17, 19-21, 44-49, 51, 53-56, 78 and 79. These claims are rejected by the Examiner using the same argumentations as for the above claims and hence the explanations hereinabove do equally apply. Therefore these claims should be allowed for the respective explanation therein.

Claim 77 is rejected as being anticipated by Starr et al. in US patent 6,807,581 (hereinafter "Starr"). Herein general comment on queues and the differences of Starr and the Application in the areas of data movement, protocol processing and queues are provided.

General Comments – Data Movement. The first difference has to do with the actual movement of data within the network interface card. According to Starr, data is moved within the INIC and evidence for that is found in multiple places. For example Starr notes that: "*Upon matching the packet summary with the CCB, assuming no exception conditions exist, the data of the packet, without network or transport layer headers, is sent by direct memory access (DMA)*

unit 68 to the destination in file cache 80 or file cache 24 denoted by the CCB." (column 8, lines 7-11). As it is noted earlier in Starr that "*When a network packet that is directed to the host 20 arrives at the INIC 22, the headers for that packet are processed by the sequencers 52 to validate the packet and create a summary or descriptor of the packet, with the summary prepended to the packet and stored in frame buffers 77 and a pointer to the packet stored in a queue.*" (column 6, lines 58-63). Hence, it is clear that there is data movement caused in Starr and clearly stated there. In another example Starr notes that "*A packet control sequencer 510 oversees the fly-by sequencer 502, examines information from the media access controller 60, counts the bytes of data, generates addresses, moves status and manages the movement of data from the assembly register 500 to SRAM 508 and eventually DRAM 512. The packet control sequencer 510 manages a buffer in SRAM 508 via SRAM controller 515, and also indicates to a DRAM controller 518 when data needs to be moved from SRAM 508 to a buffer in DRAM 512. Once data movement for the packet has been completed and all the data has been moved to the buffer in DRAM 512, the packet control sequencer 510 will move the status that has been generated in the fly-by sequencer 502 out to the SRAM 508 and to the beginning of the DRAM 512 buffer to be prepended to the packet data.*" (column 13, line 55 through column 14, line 2).

By contrast, no such data movement is required in accordance with the present invention, as specifically described with reference to the examples in Figs. 5A through 5I, and as further supported by the accompanying text of the Application. Starr notes that the advantage of the architecture proposed therein is: "*... that the packet does not, in contrast with conventional sequential software protocol processing, have to be stored, moved, copied or pulled from storage for processing each protocol layer header, offering dramatic increases in processing efficiency and savings in processing time for each packet.*" (column 15, lines 33-39). Starr specifically uses the term storage which throughout Starr refers to a storage unit and not the memory used in Starr's INIC. Starr notes that the capability to terminate TCP collect the L5 (NFS in this case) data in the INIC's local DRAM, involves the copying of the data once, from local DRAM (Frame Buffer 77) to local DRAM (INIC File Cache 80), avoiding data copies by the host software. As noted in the present Specification, translation from TCP, layer 4, to Session layer,

layer 5, is possible without locally copying the data. Hence, while Starr clearly provides an advantage over prior art of its time, it still does not resolve what is solved in the Application, namely, eliminating the need to move data within the INIC itself. Therefore the Application overcomes at least a performance limitation of Starr.

General Comments – Protocol Processing. The second difference between Starr and the present invention relates to the location of where protocol processing from layer 5 downwards is performed and controlled. In this respect the Applicants refer the Examiner to Fig. 2 of Starr which clearly notes that protocol stack 38 is part of host 20. Starr therefore notes that "*Also stored in host memory 33 is a protocol stack 38 of instructions for processing of network communications and a INIC driver 39 that communicates between the INIC 22 and the protocol stack 38.*" (column 5, lines 18-21). Starr further notes that "*In order to provide fast-path capability at the host 20, a connection is first set up with the remote host, which may include handshake, authentication and other connection initialization procedures. A communication control block (CCB) is created by the protocol stack 38 during connection initialization procedures for connection-based messages, such as typified by TCP/IP or SPX/IPX protocols... When a message, such as a file write, that corresponds to the CCB is received by the INIC, a header portion of an initial packet of the message is sent to the host 20 to be processed by the CPU 30 and protocol stack 38. This header portion sent to the host contains a session layer header for the message, which is known to begin at a certain offset of the packet, and optionally contains some data from the packet. The processing of the session layer header by a session layer of protocol stack 38 identifies the data as belonging to the file and indicates the size of the message, which are used by the file system to determine whether to cache the message data in the host file cache 24 or INIC file cache 80, and to reserve a destination for the data in the selected file cache...*" (column 7, lines 23-65). Hence, it is clear that the protocol processing is performed under the control of host 20 and that for at least every initial packet of a message within an established connection there is a need to access host 20 for certain protocol processing;

therefore, only partial relief is provided to that host. That is, every packet that is the head of a layer 5 PDU is performed by the host.

By contrast, and as can be seen in at least Figs. 4 and 6 of the present Specification, as well as the supporting text in at least paragraphs [0072] and [0083], the protocol processing is performed at the data streamer level, providing a full relief from protocol processing by the host. Specifically it should be noted that upper level protocol (ULP) processing in performed by the data streamer, a capability not found in Starr's INIC. More specifically, in an established connection in accordance with the Application there is no need to involve the host when a new message is received using the same connection. This of course not only relieves the host from processing load but also provides for a higher overall performance of the system. The differences are further noticeable from Figs. 12 and 13 of Starr that show the way that Starr communicates with the host to handle the upper layer protocol (ULP), and further in conjunction with the explanations provided by Starr in column 16 line 63 through column 17 line 36. Therefore in the Application there is no interaction between the host and the data streamer during transfer of data, in contrast with Starr.

General Comments – Queues. While queues are used in both Starr and the present invention the mere fact that both exist does not constitute a reason to assume that these have an identical function. Starr has an elaborate scheme of queues having multiple functions. Starr refers to a plurality of queues (column 29, lines 12), and specifically mentions other queues, for example, a receive queue (column 14, line 4), a free buffer queue (column 29, line 41), vector queue 2113 (column 31, line 61), and summary queue 2112 (column 31, line 67). By contrast, the Application discloses only two queues, the object queue (520) and the application queue (530). For each connection, in accordance with the invention, there is created an application queue is opened and managed until the connection is terminated. The operation of the object queue and an application queue is discussed in great detail with respect to Figs. 5A through 5I of the Specification and further in conjunction with descriptors 540. These queues are specifically designed to allow the operation of the Application in such a way that in the interface between

layer 4 and layer 5 processing there is no need to move data. It further allows processing at layer 5 of the device, as the output of the queue is a clean L4 byte stream. This functionality is explained in detail above, nor is there a need to access the host on a per massage basis. None of the queues suggested by Starr have any kind of functionality resemblance to the functionality suggested by the queues of the Application, neither is there an equivalent to the per-connection application queue shown in detail in the Application.

Claim 77. As explained above, the architecture of Starr results in data movement. Therefore, steps for data movement must be included to show the operation (see Figs. 2 and 3 of Starr). Without them, applying the steps of claim 77 on the Starr architecture would not be operative. According to the present invention, once data is moved its next movement shall only be done out of the system and onto the network. By contrast, in the movement from an ULP to a lower level protocol there is data movement according to Starr which requires memory resources and adds latency to the data movement activities. The claim is amended to more clearly address this issue and putting the claim in condition to be allowed.

Claim Rejections Under 35 USC § 103

Examiner rejects claims 6, 18, and 50 as being unpatentable over Roach in view of Starr.

Claims 6, 18 and 50. The claims should be allowed for at least being dependent claims of an allowable claim.

Examiner rejects claims 30-36, 38-43, and 60-76 as being unpatentable over Roach in view of Fishler et al. in US patent 5,954,794 (hereinafter "Fishler").

Claim 30. The claim should be allowed at least based on its dependency. Further, Fischer does not overcome the deficiency noted in Starr.

Claim 31-36, 38-43, and 60-76. These claims should be allowed at least based on its dependency.

Examiner rejects claims 20, 27, 52 and 59 as being unpatentable over Roach in view of Muller et al. in US patent 6,453,360 (hereinafter "Muller").

Claim 27 and 59. Examiner refers to Muller for the proposition that "*A pointer may be employed to identify an offset into a packet being parsed. In one embodiment, such a pointer is initially located at the beginning of the layer two protocol header. In another embodiment, however, the pointer is situated at a specific location within a particular header (e.g., immediately following the layer two destination and/or source addresses) when parsing commences. Illustratively, the pointer is incremented through the packet as the parsing procedure executes. In one alternative embodiment, however, offsets to areas of interest in the packet may be computed from one or more known or computed locations.*" (Column 25, lines 16-27). Examiner construes this to meant that the object queue of the Application points to a second descriptor if a second header has the same tuple corresponding to the first header. The only connection that is noticeable is the word *two* that may have caused the Examiner's confusion. Muller refers to a *layer two protocol* rather than on *two protocol headers*. Even if this argumentation is accepted, still nothing in Muller shows that the same tuple has to be present in both headers.

Claims 20 and 52. The claim should be allowed at least based on its dependency.

In view of the above, reconsideration and allowance of this application are now believed to be in order, and such actions are hereby solicited. If any points remain in issue which the Examiner feels may be best resolved through a personal or telephone interview, the Examiner is kindly requested to contact the undersigned at the telephone number listed below.

AMENDMENT UNDER 37 C.F.R. § 1.116
U.S. Application No. 10/014,602
Attorney Docket No. Q66695

The USPTO is directed and authorized to charge all required fees, except for the Issue Fee and the Publication Fee, to Deposit Account No. 19-4880. Please also credit any overpayments to said Deposit Account.

Respectfully submitted,

Chid S. Iyer

Chid S. Iyer
Registration No. 43,355

SUGHRUE MION, PLLC
Telephone: (202) 293-7060
Facsimile: (202) 293-7860
WASHINGTON OFFICE
23373
CUSTOMER NUMBER

Date: March 2, 2006